USAGE GUIDE
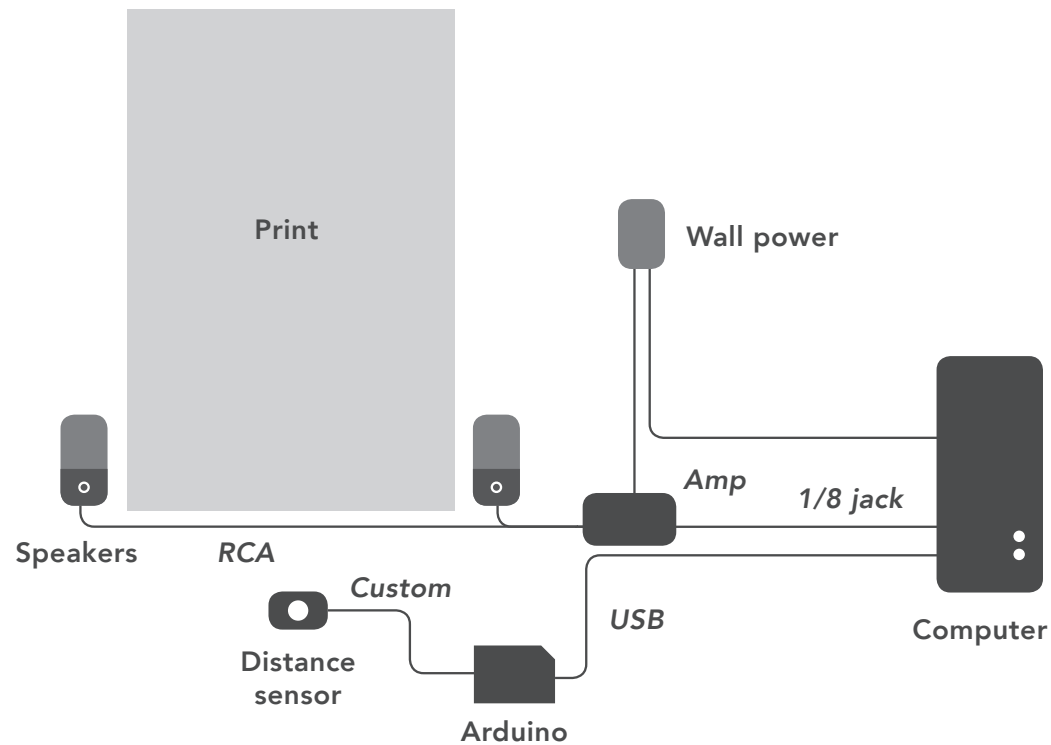
INTERACTIVE MOTION DETECTION

PAUL BUCCI
HELLO@PAULBUCCI.CA
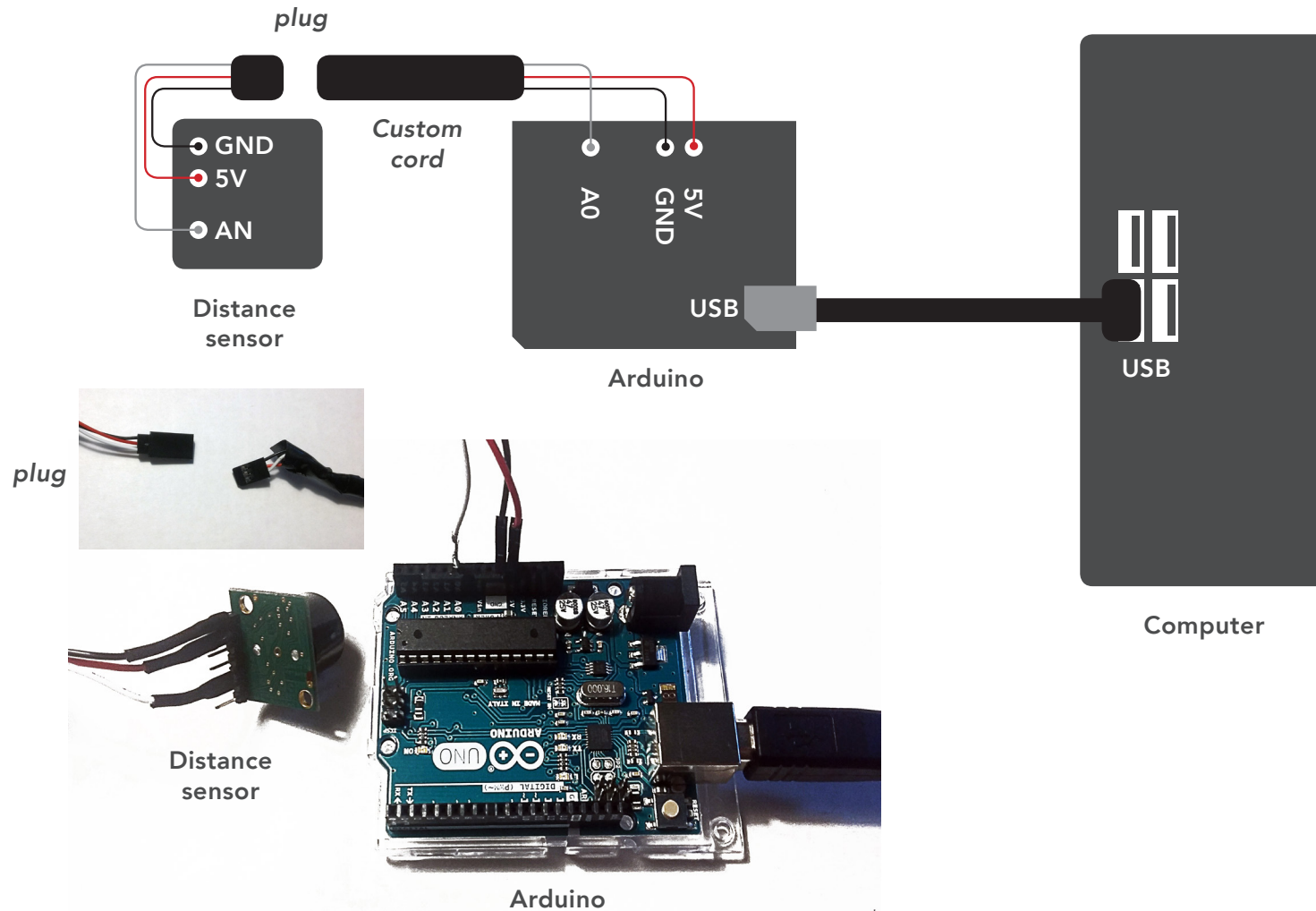604 615 3069

SETUP
CONCEPTUAL SETUP DIAGRAM
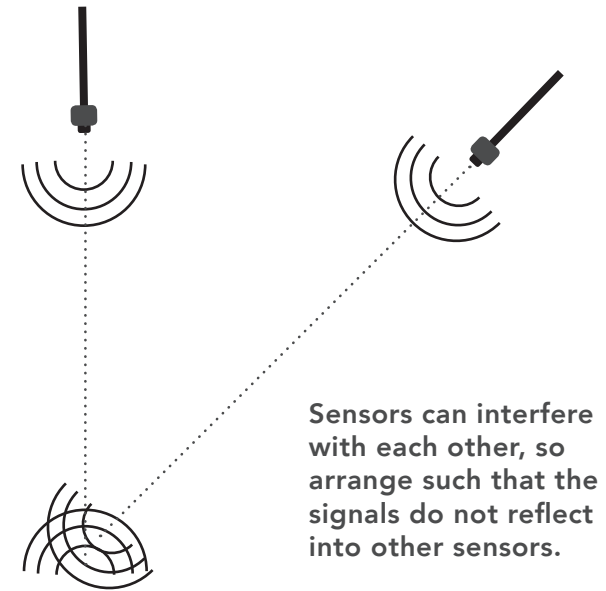
Print

Wall power

Amp

1/8 jack

Speakers

RCA

Custom

USB

Computer

Distance sensor

Arduino

SENSOR CIRCUIT DIAGRAM

plug

Custom cord

GND
5V
AN

Distance
sensor

A0

5V
GND

USB

Arduino

USB

Computer

plug

Distance
sensor

Arduino

NOTES ON ARRANGEMENT

Speaker

Sensor

Print

Distance between the end of the sensor and the body of the viewer should be less than 7 ft.

Viewer

Sensors can interfere with each other, so arrange such that the signals do not reflect into other sensors.
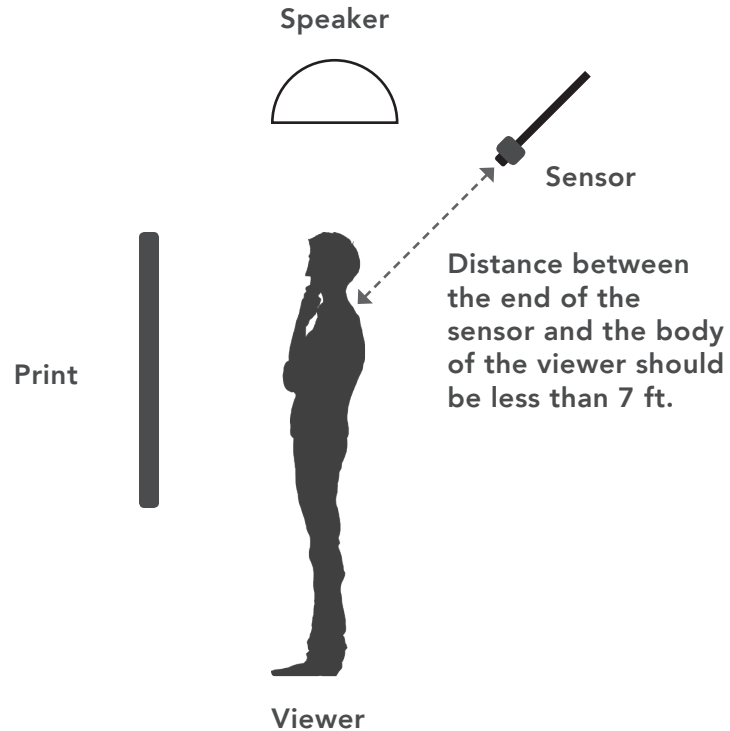
CONCEPT
TIMING THE DETECTION

The current algorithm works like this:

- If an object is detected within 7ft of the sensor, fade the song in.

- If the song is playing, and an object is detected, keep playing.

- If the song ends and an object is still detected, replay the song.

- If an object is not detected, wait 5 seconds, then fade out.

- If an object is detected within 10 seconds, fade back in from the same place in the song.

- If no objects have been detected for 10 seconds, reset the song.

The parameters of the fades/pauses/reset times can be edited in the code under the 'Options' heading.


```
                          ultrasonic_detection | Processing 3.3.4                    Java ▼

   ultrasonic_detection  ▼

38  AudioPlayer player;
39
40  // -----------------------------------------------------------------
41  // Options
42  int dataPin = 0;          // the input pin for the ultrasonic sensor
43  int delay = 5000;         // length of time the song will play for
44                            // after last trigger (ms)
45  int fadeOutLength = 2500; // length of time the fade out takes (ms)
46  int fadeInLength = 1000;  // length of time the fade in takes (ms)
47  int stopDelay = 10000;    // length of time a pause can happen before
48                            // resetting the song to beginning (ms)
49  float threshold = 84.0;   // minimum distance to trigger playback (in)
50  int minVolume = -80;      // minimum volume
51  int maxVolume = 13;       // maximum volume
52  int filterWidth = 100;    // Number of samples to take
53
54  // -----------------------------------------------------------------
55  // Program state variables
56  boolean isFading = false; // whether player is fading out

   Distance to object is 91.0.
   Distance to object is 91.0.

   ▶_ Console    ⚠ Errors
```

Editable parameters in the code

ULTRASONIC RANGE FINDING

Ultrasonic range finding works like a sonar. The device measures the amount of time it takes a sound wave to be emitted then reflected from an object. This is converted to an analog signal, i.e. a voltage, which is measured by the Arduino. The range of the sensor is roughly 8ft within a 2ft cylinder.

The biggest drawback of ultrasonic sensors is that they can interfere with each other (when one sensor sends out a signal that the other detects). To minimize interference, set them up such that they do not point at each other, nor can the signals be easily reflected into each other.

The current detection algorithm further minimizes interference by taking many samples (100/second) and finding the maximum value. This decreases the responsiveness and sensitivity of the sensor, but also decreases the chance of accidental triggers. If the sensor is not reacting quickly enough, decrease the **filterWidth** and change the filter type to **medianFilter** or **minFilter**.
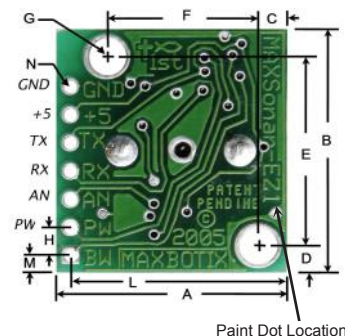
Paint Dot Location

-EZ Mechanical Dime
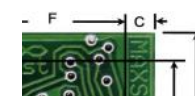
Beam size (each box is 1ft²)

```
127
128     float distance = maxFilter(samples);
129     // Uncomment this line to print sensor distance values
130     println("Distance to object is " + distance + ".");
131
```

Line to change filters.

SOFTWARE
RUNNING CONTINUOUSLY

To run the songs continuously, open the songs in a media player (such as VLC) and play on loop. Each machine has been set up with the VLC media player. Ensure that the 'Loop' button has been selected.
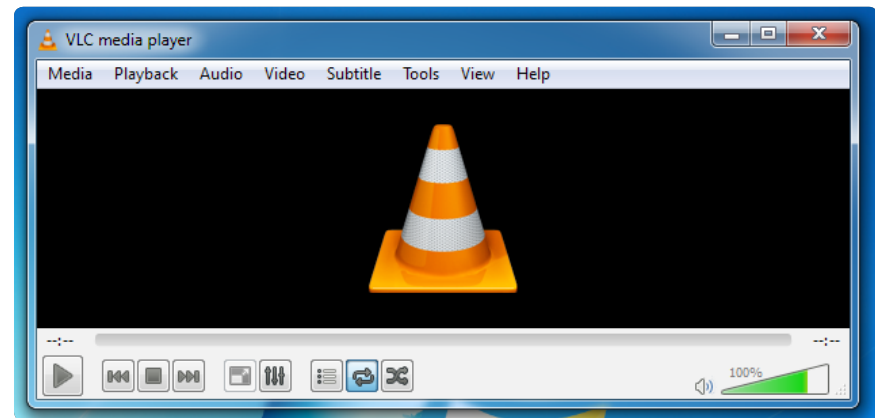
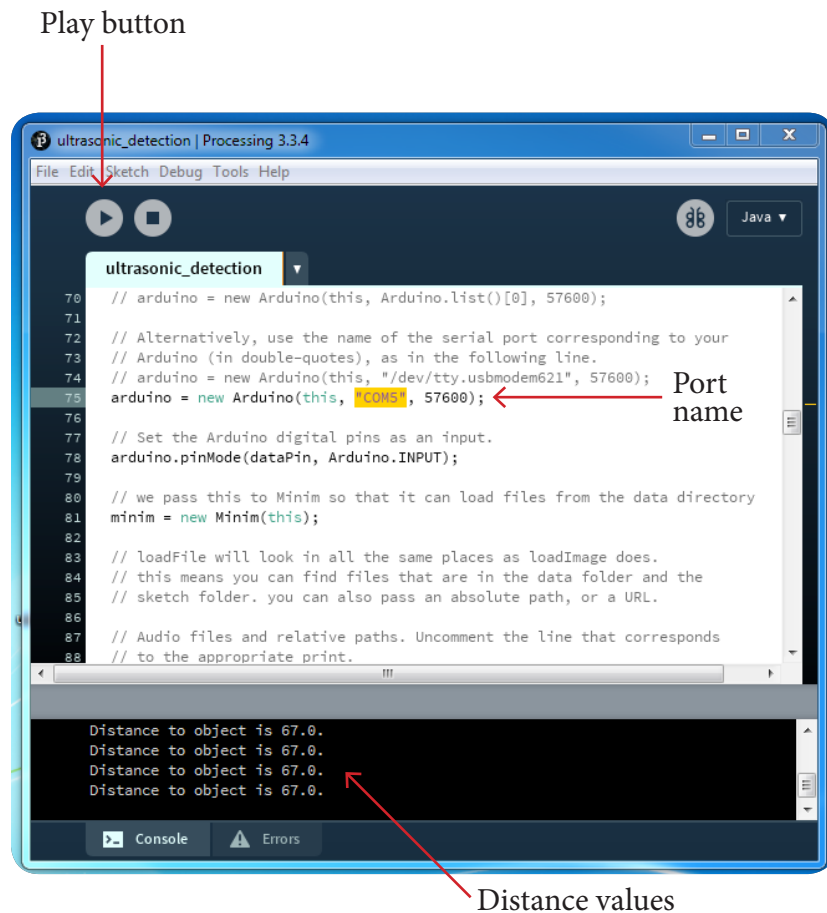The VLC player. Note the blue highlighted loop button.

SOFTWARE
RUNNING INTERACTIVE MODE

To run the interactive program:

- First open up the **Processing** program by double-clicking on **ultrasonic_detection - Shortcut**.

- You can also find this in the **ultrasonic_detection** folder in the file called **ultrasonic_detection.pde**.

- Ensure the speakers are plugged in, or the program will throw an error.

- Click on the circular play button in the top left corner.

- You should see distance values logging in the **console**. These are in **inches**. Ensure that the sensor is operating correctly by determining whether these are reasonable values.

- If there is a problem here, e.g. the distance values are unreasonably high or low, the **port name** may need to be changed. I have matched each Arduino with a computer so this shouldn't happen, but if it does, check out the attached video/below instruction to determine the correct port name, then enter in the code.

Play button

Port name

Distance values

RUNNING INTERACTIVE MODE

continued from last page

- The first thing the program logs is a list of port names. If the distance values are all very large or very small, try the next port name on the list. See video in the folder to see how to check and set port names: **ultrasonic_detection/ how_to_ check_port_names.mp4**

- The console will also log when the program resets the song.

- **Each computer should be set to a default song and clearly labeled**. However, you can change the songs by **uncommenting** the line that has the filename of the song you want to play.

- Make sure to **comment/delete** the line(s) that the songs you do not want to play.

- **Uncommenting** means deleting the two forward slashes in front of a line. The line should become colourful.

- **Commenting** means adding two slashes in the front of a line. The line should become grey.

```
90   // player = minim.loadFile("test.mp3");
91
92   // player = minim.loadFile("170608_CedarSister_ExhibitionMaster1_copy.wav");
93   player = minim.loadFile("170608_FoamWoman_ExhibitionMaster1.wav");
94   // player = minim.loadFile("170610_FoamWoman_ExhibitionMaster2_SpokenIntro.wa
95   // player = minim.loadFile("170608_Landslide_ExhibitionMaster1.wav");
96
97   }
```

Song file names

SOFTWARE
TROUBLESHOOTING

- For any problems, feel free to shoot me an email at **hello@paulbucci.ca**, or call me at **604-615-3069**.

- **If the sensor values are absurd**, ensure that you (1) have cleared the area in front of the sensor; (2) have entered the correct COM port; (3) have the ground, power, and data cables in the correct positions. If all else fails, try unplugging/replugging the Arduino and restarting the Processing program.

- **If people aren't being detected**, ensure that (1) there isn't any interference; (2) that the sensor is within 7ft of a typical person's body; (3) that the distance values are reasonable (see above); (4) amend the samples per second (i.e. **filterWidth**) or filter type (i.e. **min/max/median**; see above).

- **To tell if there's interference**, check the distance values being printed to the console. If they are rapidly changing between high and low values, there is likely interference. Try (1) repositioning the sensor; (2) changing the number of samples per second/filter method.

- **If the sensor is still too sensitive**, change the distance threshold.

- **If the song isn't playing**, ensure that the song title is correctly entered. All songs are under **ultrasonic_detection/data**.

- For more information, read the datasheets under **ultrasonic_detection/datasheets**.

- **If the program throws an error**, just close the program window and press "Play" again.

- Again, **if all else fails**, try the classic "unplug-replug-restart" fix.

- If needed, the code is stored in a private repository at **github.com/pbucci/MotionDetection**. All computers should be set up to automatically sign in to my account.